Feature-based Light Field Morphing

Zhunping Zhang, Lifeng Wang, Baining Guo,

& Heung-Yeung Shum

Microsoft Research Asia

Light Field Morphing

Light Field Morphing

A general framework for image-based 3D morphing

- Enables morphing between image-based objects
- 3D morphing without modeling
- Suitable for objects with complex surface properties (e.g., fur, subsurface scattering, hypertexture)

Related Work

- **Image morphing**
  - [Beier & Neely 92, Lee et al. 95]
- **View morphing**
  - [Seitz & Dyer 96]
- **Geometry-based 3D morphing**
  - Boundary-based (e.g., [Kent et al. 92, DeCarlo & Gallier 96, Gregory et al 99, Lee et al. 99])
  - Volume-based [Hughes 92, Lerios 95, Cohen-Or et al. 98]

Related Work

- **Plenoptic editing**
  - [Seitz & Kutulakos 98]
  - Requires 3D recons. (voxel coloring)
  - 3D recons. can be problematic
  - Restricted to diffuse scene

Morphing = Correspondence

- **Image morphing**
  = 2D pixel correspondence
- **Geometry-based 3D morphing**
  = 3D vertex correspondence
- **Light field morphing**
  = ?

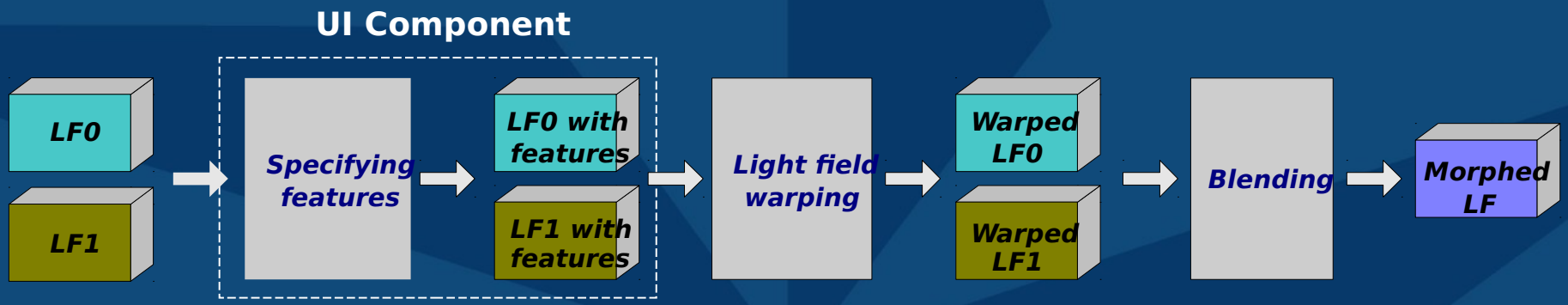
Morphing = Correspondence

- **Image morphing**
  = 2D pixel correspondence
- **Geometry-based 3D morphing**
  = 3D vertex correspondence
- **Light field morphing**
  = 4D ray correspondence

Contributions

- **A UI for specifying features in 4D ray space**
- **Ray-space warping**
  - Handling visibility changes due to object shape change

Overview

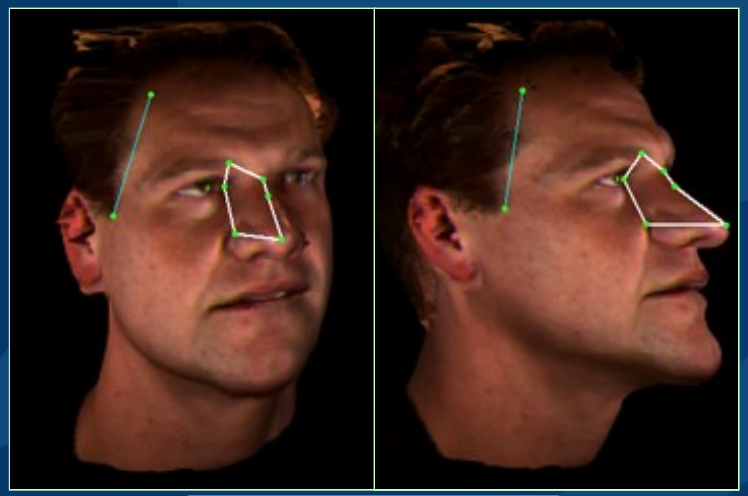

UI Component
LF0
LF1
Specifying features
LF0 with features
LF1 with features
Light field warping
Warped LF0
Warped LF1
Blending
Morphed LF

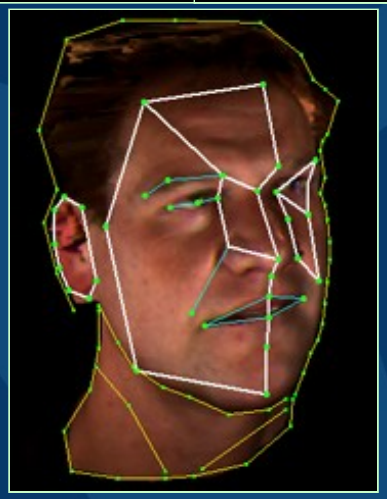
Specifying Features

- **Feature points**
  - 3D points on object surface
  - Specified by manual correspondence guided by epipolar geometry

Specifying Features

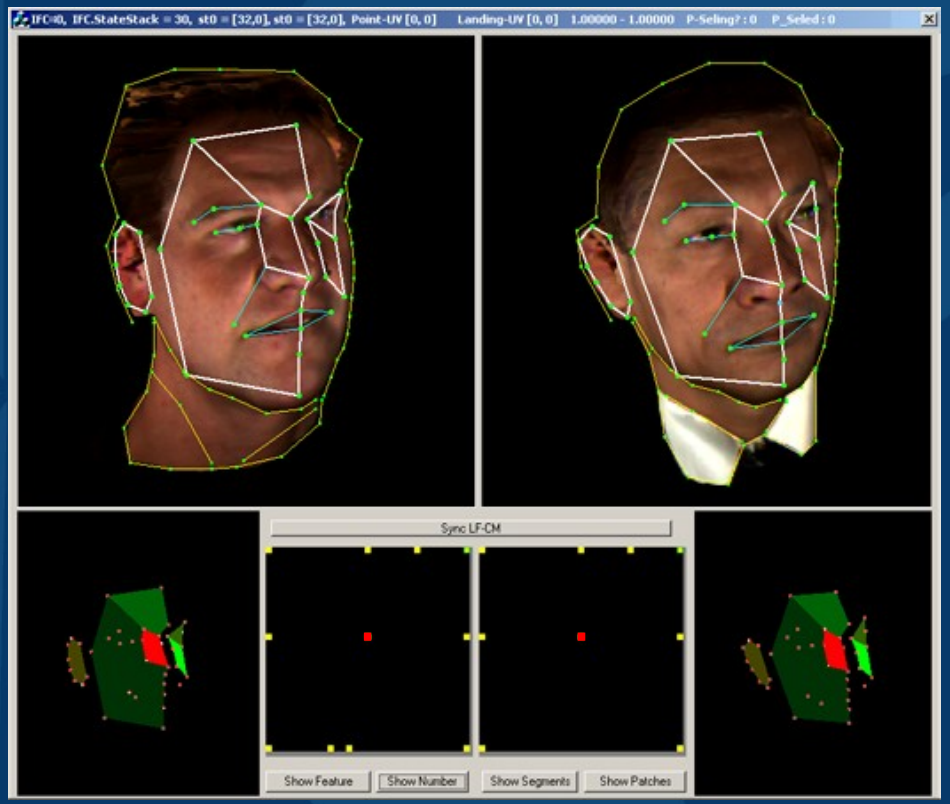
- **Feature lines**
- **Feature polygons**
  - Non-planar, but relatively flat & w/o self-occlusion
  - Necessary only in areas with visibility changes





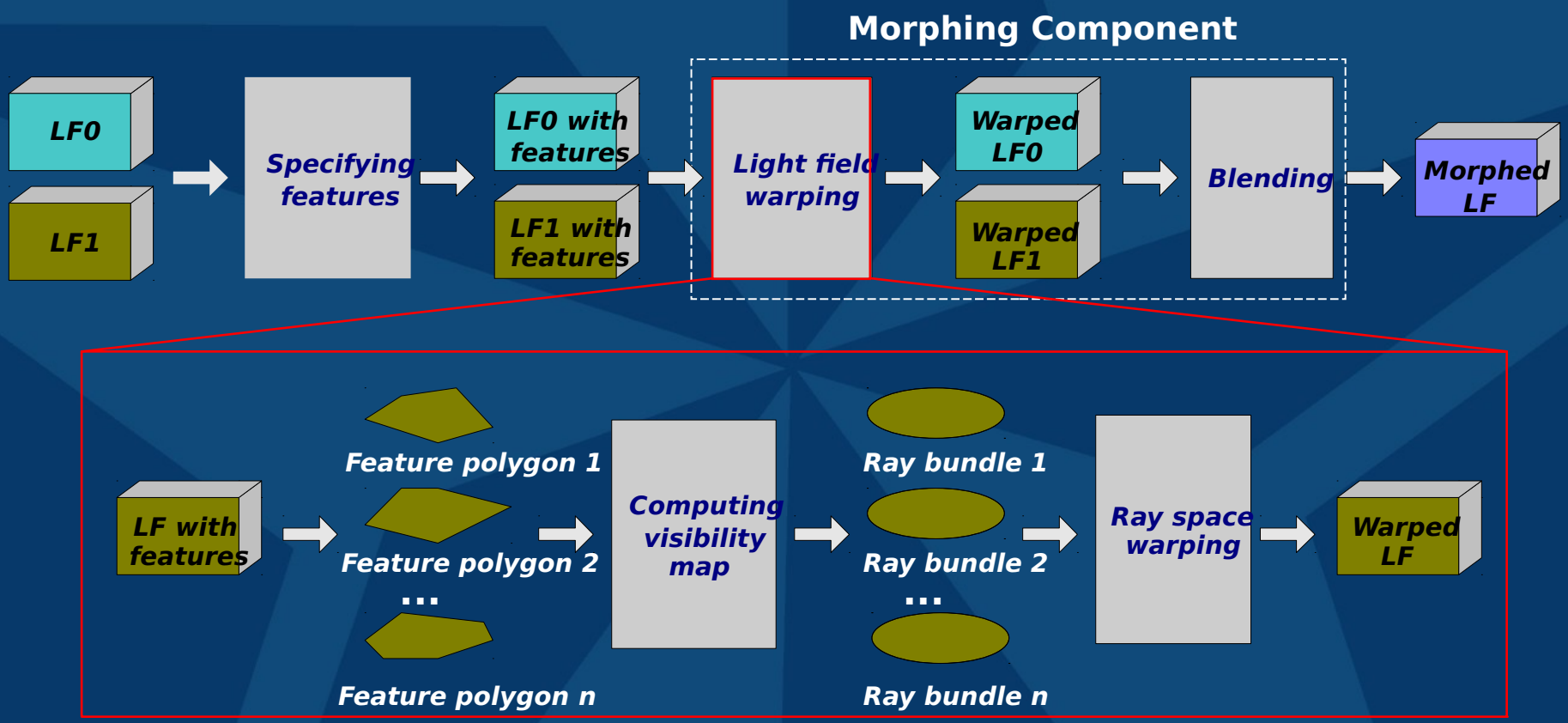
Specifying Features

- **No 3D reconstruction from feature polygons**
- **Background pixel (ray)**
  - Pixels (rays) with no visibility changes
  - Morphing controlled by background edges
- **Background edges are key-framed**

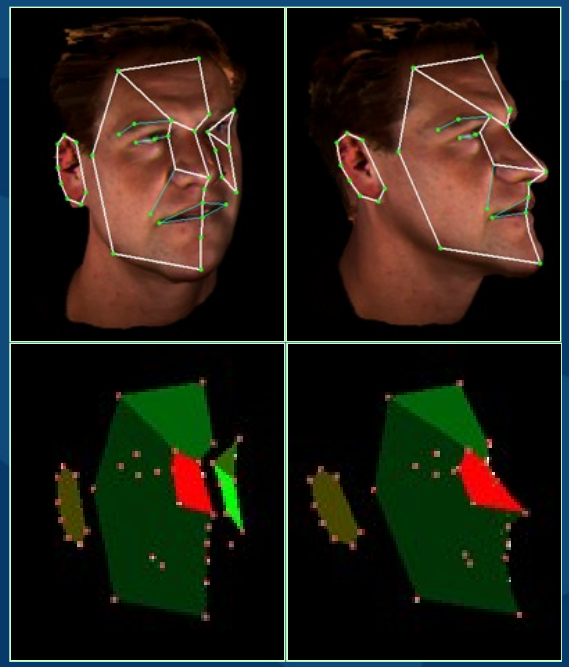


Specifying Features

Overview


Morphing Component
LF0
LF1
Specifying features
LF0 with features
LF1 with features
Light field warping
Warped LF0
Warped LF1
Blending
Morphed LF
LF with features
Feature polygon 1
Feature polygon 2
...
Feature polygon n
Computing visibility map
Ray bundle 1
Ray bundle 2
...
Ray bundle n
Ray space warping
Warped LF

Global Visibility Map



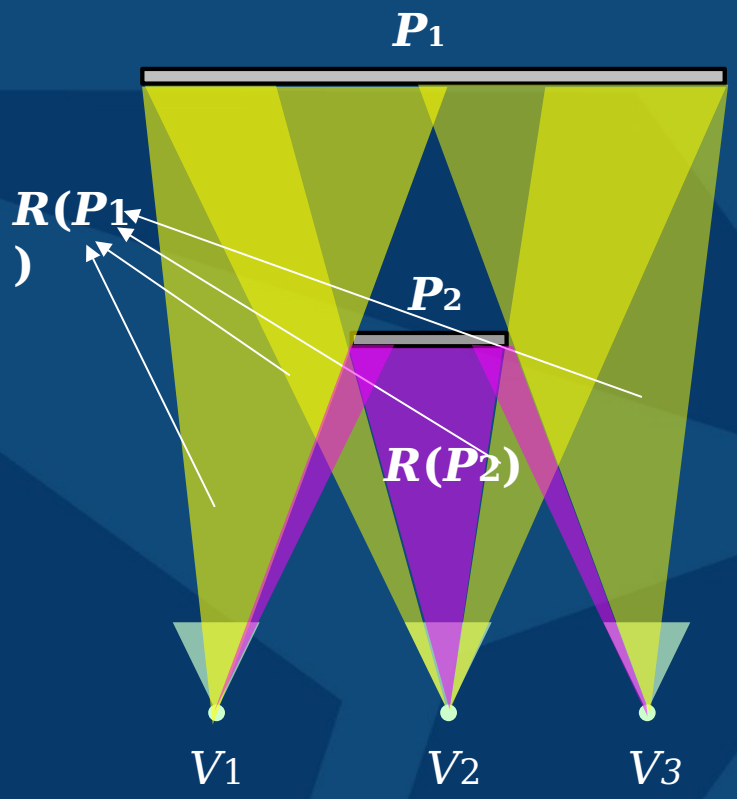
- GVM describes the visibility of feature polygons in each view
- Key to visibility processing
- GVM = A light field of false colors, associating each ray with a feature polygon

$$V(u,v,s,t) = \begin{cases} i & \text{if } rayL(u,v,s,t) \text{ belongs to } P_i \\ -1 & \text{otherwise} \end{cases}$$

Computing GVM

- **Rendering a set of non-planar but relatively flat polygons**
  - No self-occlusions
  - Two-pass OpenGL rendering with stencil buffer
- **Trade off: planar vs non-planar feature polygons**

Ray Bundles



- **Ray bundle**

  $R(P)$ = all rays associated w. feature polygon P

- **GVM decomposes light field into ray bundles**

  LF = background rays

  $+ R(P_1) + \ldots + R(P_n)$